

.....  
**Question 1 (2 points) :** Qu'est-ce que ce programme affiche ?

```
var1, var2 = 2, -3

if var2 > 0 or var1 > 0:
    var2 = var2 - 1
    var1 = var1 + 1

if var2 < 0 or var1 < 0:
    var2 = var2 + 2
    var1 = var1 + 2

if var1 and var2:
    print(var1 * var2)
else:
    print(-var1 * var2)
```

(A) (B) (C) (D)

10	-10	4	12
----	-----	---	----

**Question 2 (2 points) :** Qu'est-ce que ce programme affiche ?

```
s = 'La vie est belle'
t = s[:4]
t = t + s[-4:] + s[1]
print(t[2:2])
```

(A) (B) (C) (D)

sela	sle	selL	tela
------	-----	------	------

.....  
**Question 3 (2 points) :** Qu'est-ce que ce programme affiche ?

```
def my_func(seq, x):
    x.pop()
    seq.append(x * 2)
    return seq

s = [0, 5, 9, 11]
print(my_func(s, s[2:]))
```

(A)

(B)

(C)

(D)

**Question 4 (2 points) :** Qu'est-ce que ce programme affiche ?

```
n = 2000
cnt = 0

while n > 1:
    if n % 25:
        n = n // 2 if n % 4 else n // 4
    else:
        n = n // 25
        continue
    cnt = cnt + 1

print(cnt)
```

(A)

(B)

(C)

(D)

.....

**Question 5 (2 points) :** Qu'est-ce que ce programme affiche ?

```
s = '1011010101'
key = '010'
v = ''
for i in range(len(s)):
    if i > (len(s) - len(key)):
        break
    if s[i:i+len(key)] == key:
        v = s[i:]
        break
print(v)
```

(A)

010101

(B)

01010

(C)

01

(D)

010

**Question 6 (2 points) :** Qu'est-ce que ce programme affiche ?

```
n = 10

a = []
for i in range(n):
    a.append([])

cnt = 0
for i in range(n-1, -1, -1):
    for j in range(n):
        a[i].append(i + j)
        if a[i][j] > cnt:
            cnt = a[i][j]
print(cnt)
```

Answer: 18

**Question 7 (2 points) :** Qu'est-ce que ce programme affiche ?

```
seq = [9, 38, 7, 50, 22, 2]
seq = [seq[(i + 3) % len(seq)] for i, x in enumerate(seq)]
print(seq)
```

Answer: [50, 22, 2, 9, 38, 7]

.....  
**Question 8 (2 points) :** Combien d'étoiles ce programme affiche ?

```
def print_stars(n):  
    if n < 0:  
        return None  
    elif n < 3:  
        print('*', end = '')  
        print_stars(n-1)  
    else:  
        print(' ** ', end = '')  
        print_stars(n-2)  
  
print_stars(10)
```

Answer: 11

**Question 9 (2 points) :** Qu'est-ce que ce programme affiche ?

```
import numpy as np  
x = np.arange(16).reshape(4,4)  
x = np.flip(x, axis = 1)  
y = np.max(x, axis = 0)  
z = np.min(x, axis = 1)  
print(y - z)
```

Answer: [15 10 5 0]

**Question 10 (2 points) :** Qu'est-ce que ce programme affiche ?

```
angel = set('brunoganz')  
honey = set('ursulaadress')  
babe = set('marthekeller')  
print((honey - angel) - babe)
```

Answer: {'s', 'd'}  
Alternative correct answer: {'d', 's'}

**Question 11 (2 points) :** Qu'est-ce que ce programme affiche ?

```
s = 'Abrakadabra'  
d = {}  
for c in s:  
    if c in d.keys():  
        d[c] = d[c] + 1  
    else:  
        d[c] = 1  
print(d)
```

Answer: {'A': 1, 'b': 2, 'r': 2, 'a': 4, 'k': 1, 'd': 1}

.....

**Question 12 (6 points) :** Écrivez la fonction `count_digits` qui attend deux arguments:

- un entier positif `n` et
- une liste `d` de dix éléments, tous initialisés à zéro.

Cette fonction doit compter le nombre de fois qu'un chiffre (0, 1, 2, ..., 9) apparaît dans le nombre `n` et remplir la liste `d` en conséquence. L'élément de la liste `d` à l'indice `i` correspond au nombre d'occurrences du chiffre `i`. Pour un `n` négatif ou zéro, la fonction ne doit apporter aucune modification à la liste `d`.

Exemples:

(1) `count_digits(155027,d)` aura comme effet :

<b>indice <i>i</i></b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>d[<i>i</i>]</b>	1	1	1	0	0	2	0	1	0	0

(2) `count_digits(8285260,d)` aura comme effet :

<b>indice <i>i</i></b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>d[<i>i</i>]</b>	1	0	2	0	0	1	1	0	2	0

(3) `count_digits(1575559,d)` aura comme effet :

<b>indice <i>i</i></b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>d[<i>i</i>]</b>	0	1	0	0	0	4	0	1	0	1

**Question 13 (7 points) :** Écrivez une fonction `pearson_corr` qui attend deux arguments:

- `x`, une première liste de `n` éléments et
- `y`, une deuxième liste de `n` éléments.

Votre fonction doit retourner le coefficient de corrélation de Pearson `r`, qui se calcule à l'aide de formule suivante :

$$r = \frac{\sum_{i=1}^n ((x_i - \bar{x}) \cdot (y_i - \bar{y}))}{\sqrt{(\sum_{i=1}^n (x_i - \bar{x})^2) \cdot (\sum_{i=1}^n (y_i - \bar{y})^2)}} \tag{1}$$

où  $\bar{x}$  et  $\bar{y}$  sont les moyennes arithmétiques des listes `x` et `y`, respectivement:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \bar{y} = \frac{\sum_{i=1}^n y_i}{n} \tag{2}$$

.....